

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

NÁVRH A REALIZACE PROTOTYPU JEDNODUCHÉHO POHYBLIVÉHO ROBOTA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ MARKO

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

NÁVRH A REALIZACE PROTOTYPU JEDNODUCHÉHO POHYBLIVÉHO ROBOTA

DESIGN AND REALIZATION OF A PROTOTYPE OF A SIMPLE MOBILE ROBOT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ MARKO

VEDOUcí PRÁCE

SUPERVISOR

Ing. STRNADEL JOSEF, Ph.D.

BRNO 2008

Abstrakt

Táto bakalárska práca popisuje návrh a implementáciu jednoduchého pohyblivého robota. Prvá časť je zameraná na teoretické znalosti potrebné na konštrukciu jednoduchého robota. V druhej časti je načrtnuté moje riešenie. Okrem mechanickej časti je popísaná tiež programová časť riadenia robota. V tretej časti sú uvedené praktické skúsenosti z testovania prototypu robota.

Klíčová slova

robot, krokový motor, servo pohon, automatizovaný podvozok

Abstract

This bachelor's thesis describes designation and implementation of simple mobile robot. The first part is focused on theoretic knowledge needed for construction of simple robot. There is shown my solution of problem in the second part. There is also described software aspect of robot aside from mechanical construction in this chapter. In the third part, there are mentioned practical experiences of testing of robot prototype. part.

Keywords

robot, stepper motor, servo drive, automated support frame

Citace

Juraj Marko: Návrh a realizace prototypu jednoduchého pohyblivého robota, bakalářská práce, Brno, FIT VUT v Brně, 2008

Návrh a realizace prototypu jednoduchého pohyblivého robota

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Josefa Strnadela, Ing., Ph.D.

.....

Juraj Marko

14. mája 2008

© Juraj Marko, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

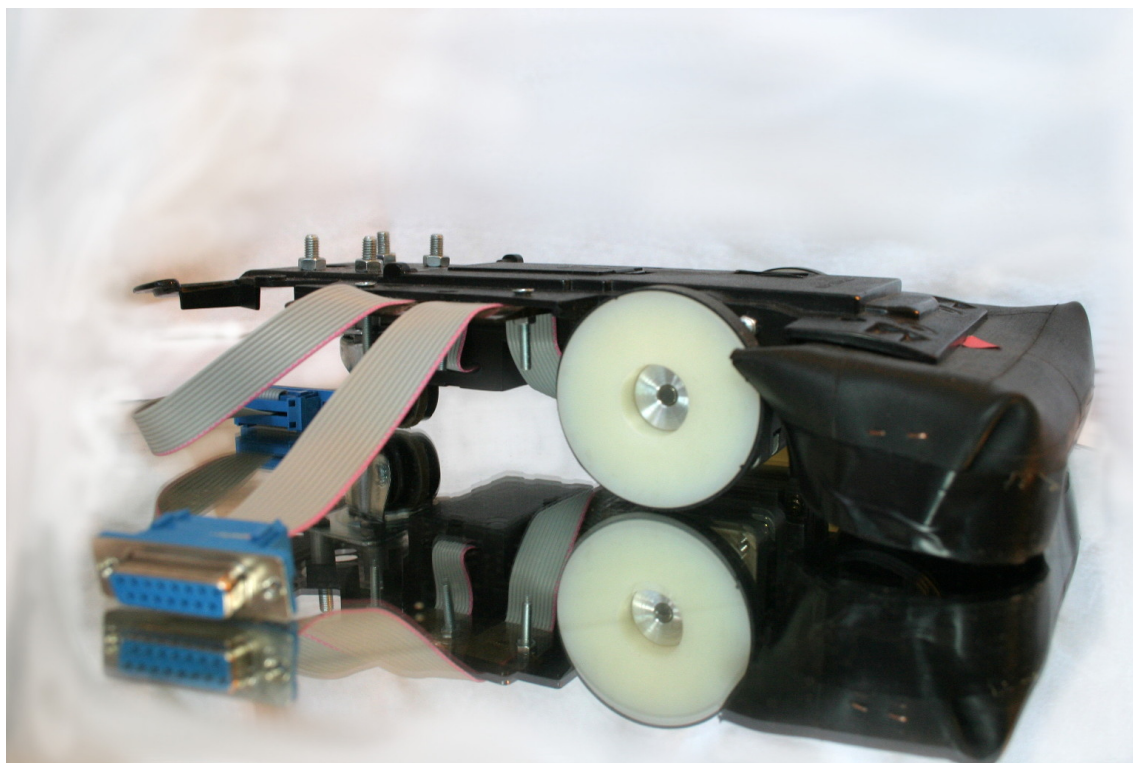
1	Úvod	3
1.1	Čo je to robot?	3
1.1.1	Môj mobilný robot	4
1.2	Motivácia	4
1.3	Členenie práce	4
2	Teória	5
2.1	Pohonná jednotka	5
2.1.1	Servopohon	5
2.1.2	Krokový motor	6
2.2	Senzory	9
2.2.1	Kontaktné senzory	9
2.2.2	Bezkontaktné senzory	9
2.2.3	Snímanie čiernej čiary	10
2.3	Mechanika	11
2.3.1	Jednokolesové podvozky	12
2.3.2	Dvojkolesové podvozky	12
2.3.3	Trojkoľosové s štvorkolesové podvozky	12
2.3.4	Viackoľosové podvozky	13
2.3.5	Diferenčne riadené podvozky	13
2.4	Riadenie	14
2.4.1	FITkit	15
2.4.2	Pda	15
2.4.3	PC	15
3	Návrh a implementácia	16
3.1	Pohonná jednotka	16
3.1.1	Zvolený motor	16
3.1.2	Budenie motoru	16
3.2	Senzory	17
3.2.1	Zvolený senzor	17
3.2.2	Rozloženie senzorov	17
3.3	Mechanika	18
3.3.1	Tretie voľne otočné koliesko	18
3.4	Riadenie	18
3.4.1	Pripojovací kábel	18
3.4.2	Knižnica robotlib	18
3.4.3	Riadiaci program	22

4	Testovanie	25
4.1	Priebežne testovanie	25
4.1.1	Testovanie riadenia jedného motoru	25
4.1.2	Testovanie podvozku po osadení motorov a tretieho oporného kolesa	25
4.1.3	Testovanie snímačov	26
4.2	Testovanie kompletného robota	27
4.2.1	Testovanie jednoduchej jazdy bez navigácie	27
4.2.2	Testovanie s navigáciou	28
4.2.3	Výsledky testovania	28
5	Záver	29
5.1	praktické využitie	29
A	Fotografie robota	31

Kapitola 1

Úvod

Zadaním mojej práce bolo navrhnuť a vyrobiť prototyp jednoduchého mobilného robota. Prvou vecou, na ktorú si potrebujeme odpovedať je, čo je to robot. Väčšina ľudí si pod pojmom robot predstaví humanoidného robota. V skutočnosti je väčšina robotov úplne inej konštrukcie.



Obrázok 1.1: hotový robot zaparkovaný na zrkadle

1.1 Čo je to robot?

Robot je automatické zariadenie schopné reagovať na podnety okolia a na toto okolie spätne pôsobiť [10], kde sa uvádza aj:

Slovo robot údajne po prvýkrát použil český spisovateľ Karel Čapek vo svojom diele R.U.R.. Pôvodne chcel použiť slovo Labor ale na radu svojho brata sa nakoniec priklonil k slovu robot, ktorým chcel zdôrazniť, že roboty sú určené pre vykonávanie ťažkej a úmornej práce (čiže roboty). V skutočnosti bolo slovo robot používané už v sedemnástom storočí, v rakúskej nemčine, vo význame „otrocká práca poddaných“.

Z definície vidíme, že robot je v podstate ľubovoľné automaticky pracujúce zariadenie. Toto zariadenie je schopné nejakým spôsobom vnímať podnety zo svojho okolia (senzormi), a je schopné na okolie vplývať (napríklad tým, že sa po ňom presúva).

V niektorých prípadoch sa slovom robot označujú aj virtuálne počítačové automaty vykonávajúce nejaké opakujúce sa činnosti, ktoré potrebujú isté inteligentné riadenie. Príkladom takýchto robotov sú vyhľadávacie roboty, ktoré používajú rôzne vyhľadávače na indexovanie internetových stránok.

1.1.1 Môj mobilný robot

Môj robot je v podstate automatizovaný podvozok schopný snímať čiary na podlahe ktorými je naznačená jeho trasa. Okrem toho je schopný detekovať križovatku takýchto čiar a podľa zadania štartovej a cieľovej pozície sa na základe informácií o topológii trasy je schopný určiť smer, ktorým bude pokračovať za križovatkou.

1.2 Motivácia

Mobilného robota je možné použiť napríklad vo fabrikách ako automatizovanú lokomotívu pre rozvoz materiálu. Výhodou je proti dopravným pásom alebo len automatizovanej železnici, že dráhu môžeme veľmi jednoducho prispôbovať vyvíjajúcim sa potrebám fabriky. To je veľmi dôležitá vlastnosť obzvlášť, keď sú na fabriky kladené veľké nároky v oblasti flexibility.

1.3 Členenie práce

V prvej kapitole sa budem venovať teoretickým informáciám z oblastí, do ktorých som pri mojom návrhu nahliadol. Nakoľko oblasti, do ktorých návrh mobilného robota zasahuje je veľa a táto publikácia má obmedzený rozsah, budem sa týmito областiam venovať len stručne. V tejto kapitole tiež spomeniem niektoré základné výhody a nevýhody pre moje riešenie.

V druhej kapitole sa budem venovať mnou zvolenému návrhu a konštrukcii robota. Podrobnejšie popíšem výhody a nevýhody zvolených a vylúčených riešení. Tiež popíšem riadiaci program môjho robota a knižnicu, ktorú k tomu používa.

V ďalšej kapitole sa budem venovať testovaniu jednotlivých častí robota počas vývoja a na záver aj kompletne zhotoveného robota.

Kapitola 2

Teória

Robota ako celok je možné rozdeliť na niekoľko jednoduchších častí. V tejto kapitole sa budem potsupne venovať možnostiam, ako navrhnuť, zhotoviť, alebo zvoliť nasledujúce časti robota:

- pohonná jednotka,
- senzory,
- mechanika,
- riadenie.

2.1 Pohonná jednotka

Bežne sú pri výrobe mobilných robotov používané 2 druhy pohonných jednotiek. Servo pohon a krokový motor.

2.1.1 Servopohon

Servopohony sa skladajú zo samotného motoru (obvykle elektrického) a z riadiacej jednotky so spätnou väzbou. Môže to byť buď senzor a regulátor rýchlosti, ktorý sa používa u rýchlostného serva, alebo senzor a regulátor polohy u polohového serva.[13] Senzor nám umožňuje udržiavať servo v želanej rýchlosti alebo polohe, podľa druhu použitého servopohonu.

Polohový servopohon (obrázok 2.1) je možné používať pri kráčajúcich robotoch na otáčanie nôh v jednotlivých kĺboch. Tiež sa používa na ovládanie natočenia riadenej nápravy pri robotoch s riadenou nápravou.

Rýchlostné servopohony sa používajú na pohon kolesových a pásových robotov. Umožňujú nám riadenie rýchlosti posunu robota. Nevýhodou je pomerne nepresné riadenie a meranie prejdenej vzdialenosti.

V priemysle sa servopohony často používajú na manipuláciu s ventilmi, stavidlami, vetrákmi, na pohon robotických ramien, na posun noža v CNC obrábacích strojoch (obrázok 2.2) a podobne.

Riadenie servopohonu je realizované dĺžkou ovládacieho impulzu, ktorý mení uhol natočenia polohového servopohonu alebo rýchlosť otáčania rýchlostného servopohonu. Výhodou riadenia pomocou dĺžky impulzu je jednoduchosť pri ovládaní pomocou mikrokontroléru.



Obrázok 2.1: modelársky polohový servopohon
zdroj: <http://geekhobbies.com/the-parts-of-a-remote-control-aircraft/>

Nevýhodou je, že pomocou počítača s operačným systémom, ktorý nie je určený pre aplikácie v reálnom čase, sa ťažko generujú signály presnej dĺžky.

2.1.2 Krokový motor

Krokový motor nám umožňuje presné polohovanie. Neposkytuje informáciu o vykonaní požadovaného natočenia. Často sa používajú v aplikáciách, kde je potrebné presne stanoviť posunutie, napríklad vystavenie čítacej hlavy pevného disku či disketovej mechaniky (obrázky 2.3 a 2.4). Problém môže nastať v prípade, že motoru v pohybe bráni nejaká prekážka. Vtedy dochádza k strate informácie o presnej polohe. Existujú dva základné druhy krokových motorov:

- Unipolárne krokové motory
- Bipolárne krokové motory

Okrem unipolárnosti a bipolárnosti sa môžu krokové motory od seba líšiť počtom krokov, ktoré je potrebné vykonať na otáčku. Čím je počet krokov vyšší, tým je motor presnejší, nevýhodou je, že pri vysokom počte krokov je potrebné na dosiahnutie vyšších otáčok motoru rýchle striedanie riadiacich impulzov. Ďalej sa môžu motory líšiť počtom vinutí. Najbežnejší počet je 4-5 vinutí, pri malých motoroch.

Unipolárne krokové motory majú jeden pól vinutí vyvedený na spoločný vodič. Opačný koniec vinutí je vyvedený ako jednotlivé fázy. Spoločný napájací vodič je väčšinou pripojený na kladný pól zdroja a jednotlivé fázy sú pripojené k zemi pomocou spínačov.[6] Fázy motoru sú postupne uzemňované spínaním spínačov podľa zvoleného ovládacieho algoritmu krokového motoru. Existujú 3 základné metódy riadenia unipolárneho krokového motoru:

- Unipolárne jednofázové riadenie s plným krokom
- Unipolárne dvojfázové riadenie s plným krokom
- Unipolárne riadenie s polovičným krokom



Obrázok 2.2: servopohony v priemysle: CNC fréza
zdroj: <http://www.industrialhobbies.com/main.htm>

Unipolárne jednofázové riadenie s plným krokom je najjednoduchší spôsob riadenia krokového motoru. Prúd prechádza v každom okamihu práve jednou cievkou motoru (obrázok 2.5). Výhodou je malý odber, ale nevýhodou je nízky krútiaci moment motoru. Výhodou tohto riešenia je jednoduché zapojenie riadiacej elektroniky. Pre menšie motory stačí použiť integrovaný obvod ULN2803. V jeho puzdre je dostatok budičov pre ovládanie dvoch krokových motorov so 4 vinutiami.[7]

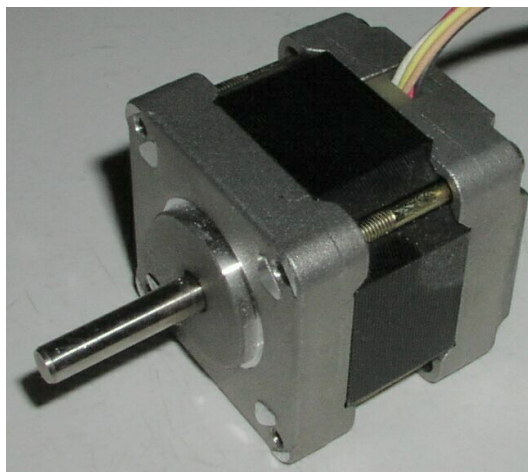
Unipolárne dvojfázové riadenie s plným krokom je o niečo komplikovanejšie ako jednofázové. Zapojenie riadiacej elektroniky síce zostáva zhodné, ale musíme pomocou riadiaceho programu uzemniť vždy dve susedné vinutia (obrázok 2.6). Nevýhodou je dvojnásobný odber elektrického prúdu a výhodou dvojnásobný krútiaci moment.

Striedaním unipolárneho jednofázového a dvojfázového riadenia s plným krokom dosiahneme **Unipolárne riadenie s polovičným krokom**. Riadenie s *plným krokom* znamená, že na otočenie rotoru o jednu otáčku treba presne toľko krokov, koľko zubov má stator použitého motoru. Riadenie s *polovičným krokom* znamená, že na otočenie o jednu otáčku potrebujeme dvojnásobný počet krokov (obrázok 2.7) a tým dosiahneme dvakrát väčšiu presnosť.[7]

Bipolárne krokové motory majú vyvedené všetky vývody cievok osobitne (nemajú zlúčené kladné póly do jedného vývodu). To nám umožňuje nabudiť vždy dve protiľahlé cievky opačnými magnetickými pólmi a tým zdvojnásobiť krútiaci moment motoru.[7] Nevýhodou je väčší odber elektrického prúdu, komplikovanejšie zapojenie ovládacej logiky, väčší počet vývodov a nakoniec aj komplikovanejší riadiaci program. Existujú 3 základné metódy riadenia bipolárneho krokového motoru:

- Bipolárne jednofázové riadenie s plným krokom
- Bipolárne dvojfázové riadenie s plným krokom
- Bipolárne riadenie s polovičným krokom

Postupy pri týchto riadeniach sú obdobné ako pri unipolárnom riadení, len s tým rozdielom, že vždy dve protiľahlé cievky sú nabudené opačne.

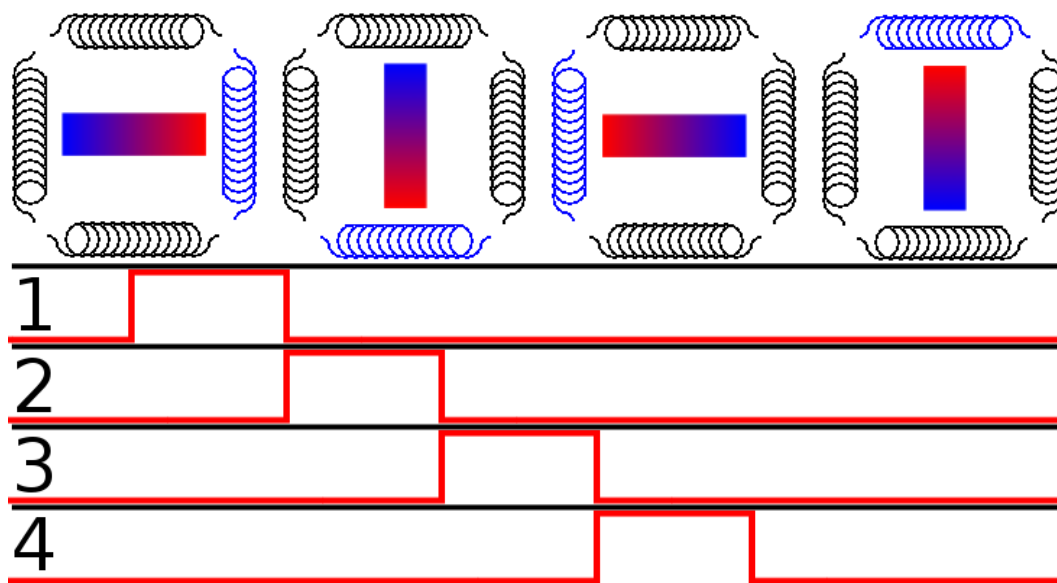


Obrázok 2.3: krokový motor
zdroj: <http://robotika.cz/articles/steppers/cs>



Obrázok 2.4: stator a rotor krokového motoru
zdroj: <http://robotika.cz/articles/steppers/cs>

Smer otáčania motoru ovládame tým, či prejdeme do nasledujúceho kroku, alebo do predchádzajúceho kroku. Na riadenie postačuje jednoduchý stavový automat, v ktorom máme uložený aktuálny stav krokového motoru a sme schopní generovať stav nasledujúci, alebo predchádzajúci a tým riadiť smer otáčania krokového motoru. Rýchlosť otáčania krokového motoru sa riadi rýchlosťou striedania riadiacich impulzov. Pri dosiahnutí maximálnej rýchlosti motor nestihne včas vykonať posunutie o jeden krok a ďalšie zmeny impulzu spôsobujú kmitanie motoru na jednom mieste. Tesne pred dosiahnutím maximálnych otáčok motoru prudko klesá krútiaci moment. Pri otáčaní krokového motoru na nízkych otáčkach je možné pozorovať trhaný chod motoru. Motor poskočí o krok do novej polohy, v nej zastane a zotrúva tak až do vygenerovania nasledujúceho impulzu. Zvyšovaním otáčok sa chod motoru stáva pravidelnejším.



Obrázok 2.5: unipolárne jednofázové riadenie s plným krokom

2.2 Sensory

Senzor je zariadenie registrujúce fyzikálnu veličinu a meniac ju na elektrický signál [12]. Senzor je pre robota zariadenie, ktoré mu umožňuje vnímať okolitý svet. V robotike môžeme senzory rozdeliť do dvoch základných skupín:

- Kontaktné senzory
- Bezkontaktné senzory

2.2.1 Kontaktné senzory

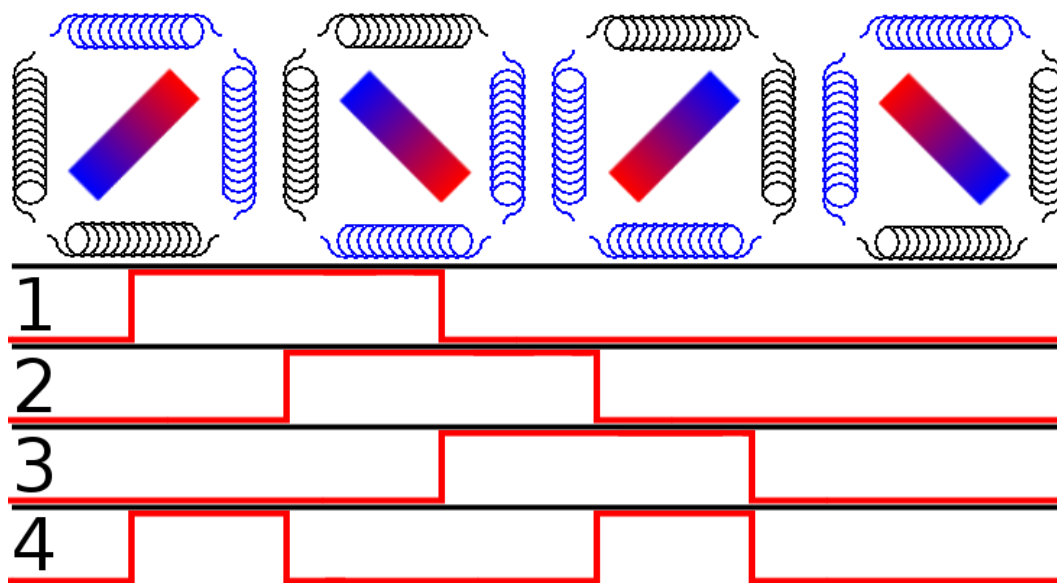
Kontaktné senzory sú senzory, ktoré prichádzajú pri meraní do priameho kontaktu s objektom, na ktorom je meranie vykonávané.

V robotike je v domácich podmienkach najbežnejším kontaktným senzorom spínač. Pomocou spínača použitého ako nárazník môže robot detekovať stretnutie s prekážkou. Tiež je možné spínače použiť na detekovanie okraja „stola“, zdvihnutia robota zo zeme, preťaženie nákladom a podobne.

Ďalšími senzormi sú teplomery, ktoré nám umožňujú snímanie teploty okolia alebo nejakého predmetu, prípadne aj niektorých častí robota (detekovať prehriatie motorov a podobne). Existujú tiež snímače umožňujúce detekovať výskyt určitej chemickej látky, napríklad vo vzduchu a tak detekovať nebezpečenstvo otravy, vyhľadávať bomby... Kontaktné sa dá merať aj pevnosť povrchu, drsnosť... rovnosť...

2.2.2 Bezkontaktné senzory

Bezkontaktné senzory nám umožňujú meranie fyzikálnych veličín bez priameho kontaktu s meraným objektom. Sú to napríklad ultrazvukové detektory prekážok, laserové merače vzdialeností, kamery, radary, mikrofóny, bezkontaktné teplomery, infračervené snímanie prekážok a kontrastných značiek a podobne a hlavne indukčné senzory.



Obrázok 2.6: unipolárne dvojfázové riadenie s plným krokom

Podstatnou výhodou bezkontaktných snímačov je schopnosť zmerať fyzikálnu veličinu bez nutnosti priameho kontaktu s meraným objektom. To umožňuje napríklad detekovať prekážku skôr, ako robot do prekážky narazí. To môže byť vhodné napríklad v aplikáciách, kde by zrážka spôsobovala poškodenia na strane robota (ľahký robot pohybujúci sa značnou rýchlosťou), alebo na strane okolia (kolízia ťažkého robota s človekom). Tiež nám to umožňuje lepšie plánovanie obchádzania prekážok.

Bezkontaktné snímače nám umožňujú snímanie vizuálnych značiek v okolí (napríklad na stenách a podlahe). To môže byť využité u robota, ktorý má jazdiť podľa kontrastnej čiary na podlahe.

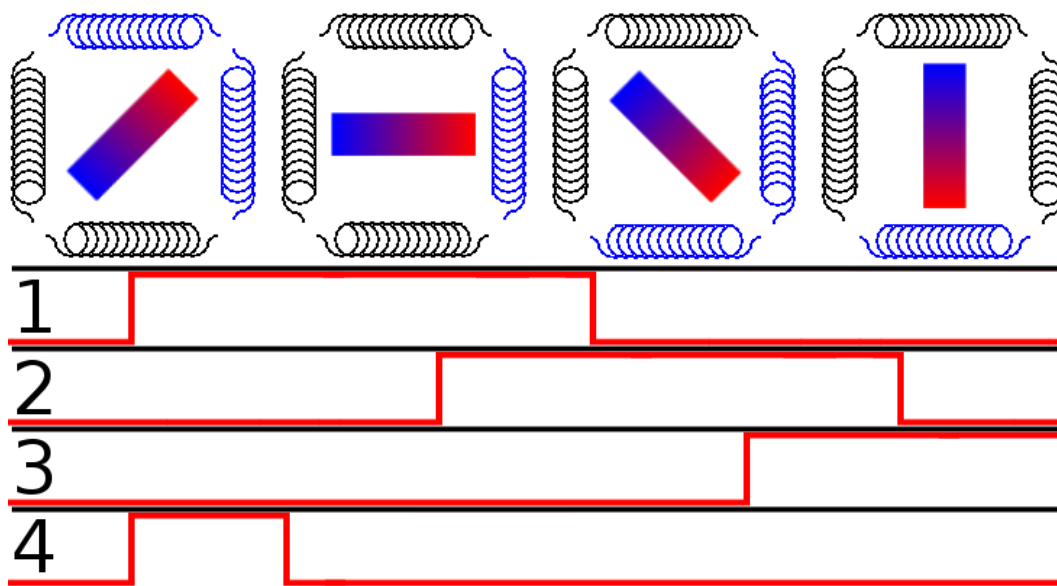
Zaujímavou vecou je možnosť detekovania nízkonapäťovej indukčnej slučky napríklad pri vyhľadávaní okrajov pracovnej plochy a návratovej cesty.[11]

2.2.3 Snímanie čiernej čiary

Bežný navigačným prvkom (v amatérskych podmienkach) pre roboty býva čierna čiara na bielom podklade. Existuje niekoľko spôsobov ako čiaru detekovať.

Jednoduché je snímanie pomocou infradiódy a infrafoto tranzistoru alebo laseru a fotodiódy. Nevýhodou tohto riešenia je len informácia o tom, či sa nachádza pod senzorom tmavá alebo svetlá oblasť. Nedostaneme žiadnu informáciu o polohe, hrúbke a smere čiary. Preto je nutné inštalovať niekoľko snímačov v závislosti od toho, ako presnú informáciu o čiare potrebujeme. Snímač si môžeme pomerne jednoducho zhotoviť sami, ale existujú tiež priemyselné snímače, ktoré vyrábajú špecializované firmy. Tie nachádzajú uplatnenie v automatizácii a riadení v závodoch, ale tiež v robotike.[9]

Komplikovanejšou možnosťou je použitie jednoriadkového[1] alebo viacriadkového ccd snímača. Načítanie informácie z ccd snímača je síce náročnejšie, ale dostaneme presnejšiu informáciu o farbe, hrúbke, polohe a v prípade použitia viacriadkového snímača aj o smere detekovanej čiary. Je možné tiež rozoznávať čiary podľa farieb, pri použití farebného ccd snímača, lebo pri použití niekoľkých snímačov s príslušnými farebnými filtermi.



Obrázok 2.7: unipolárne riadenie s polovičným krokom

Pri vlastnom návrhu snímača vychádzame z jednoduchej vlastnosti svetla. Svetlo určitej vlnovej dĺžky sa odráža od určitého povrchu. Tento fakt ľudské oko vníma ako farbu povrchu. Napríklad list modrého papiera vnímame ako modrý, lebo atómy jeho povrchu pohltili všetky vlnové dĺžky viditeľného spektra svetla okrem vlnovej dĺžky, ktorú ľudské oko vníma ako modrú. Biely povrch odráža celé viditeľné spektrum a naopak čierny povrch celé viditeľné spektrum pohlcuje.[14] Podobne sa chovajú povrchy aj k neviditeľným oblastiam spektra. Napríklad infračervené svetlo sa odráža od bieleho povrchu a naopak matný čierny povrch infračervené svetlo pohlcuje. Na tomto princípe sa dá zhotoviť jednoduchý senzor zložený z infradiódy a infrafototranzistoru. Infrafototranzistor reaguje na infražiarenie, ktoré z infradiódy dopadá na zem a od bieleho podkladu sa odráža do infrafototranzistrou, alebo sa neodráži od čiernej čiary.[4]

2.3 Mechanika

Po mechanickej stránke sa roboty výrazne odlišujú. Existujú humanoidné roboty, robotické ramená, lietajúce roboty, robotické podvozky a mnoho iných druhov robotov. Ja sa v tejto časti budem venovať mechanike robotických podvozkov. Tie ale môžeme rozdeliť tiež na niekoľko podskupín[8].

- Kráčavé robotické podvozky
- Kolesové robotické podvozky
- Pásové robotické podvozky
- Plazivé robotické podvozky

Vzhľadom k rozsahu tejto práce sa budem ďalej venovať len kolesovým robotickým podvozkom. Kolesové robotické podvozky je možné deliť podľa počtu a usporiadania kolies na niekoľko základných skupín.

- Jednokolesové podvozky
- Dvojkolesové podvozky
- Trojkolesové a štvorkolesové podvozky
- Viackolesové podvozky
- Diferenčne riadené podvozky

2.3.1 Jednokolesové podvozky

Ide o roboty využívané momentálne výhradne na laboratórny výskum. Ich konštrukcia je unikátna tým, že celý robot sa nachádza vo vnútri jedného kolesa. To je síce staticky nestabilné (jeho rovnováhu udržiava len gyroskop), ale už pri miernej rýchlosti sa jeho stabilita zvyšuje. Zatačanie je realizované pomocou naklonenia kolesa do strany. To sa využíva aj pri jazde po naklonenej rovine. Výhodou je, že robot má len jeden kontakt so zemou a nemá žiaden rám a tak nemôže ostať „visieť“ pri prejazde cez terénne nerovnosti. Nevýhodou je veľká spotreba gyroskopu.[5]

2.3.2 Dvojkolesové podvozky

Podobne ako jednokolesové podvozky musia aktívne udržiavať stabilnú polohu. Využívajú sa prevažne na laboratórny výskum. Z hľadiska usporiadania kolies môžeme tieto roboty rozdeliť na dva typy.

- Dvojkolesové roboty so sériovým usporiadaním kolies
- Dvojkolesové roboty s paralelným usporiadaním kolies

Na udržiavanie správnej polohy môže robot používať gyroskop, alebo môže fungovať ako reverzné kyvadlo.[3]

2.3.3 Trojkolesové s štvorkolesové podvozky

Trojkolesové a štvorkolesové podvozky sú najväčšou skupinou mobilných kolesových robotov. Tieto podvozky sú staticky aj dynamicky stabilné a preto u nich nie je potrebné realizovať stabilizáciu gyroskopom.[15] Troj a štvor kolesové podvozky je možné riadiť jednak diferenčne a jednak riadeným kolesom či nápravou. Diferenčne riadeným podvozkom sa budem venovať neskôr. V tejto časti budem rozoberať len podvozky s riadenou nápravou alebo kolesom.

Pre jednoduchosť uvažujme štvorkolesového robota. Podobné princípy budú platiť aj pre trojkolesového. Robot môže mať niekoľko spôsobov riadenia a pohonu. Po konštrukčnej aj po riadiacej stránke je najjednoduchšia jedna hnacia a jedna riadiaca náprava. Mechanicky o niečo zložitejšie (hlavne pri štvorkolesovom robote) je usporiadanie, kde je hnaná riadiaca náprava. Druhá náprava je voľná. Z hľadiska riadenia sú obe riešenia totožné. Komplikované riešenie je náhon a riadenie všetkých kolies. To zvyšuje manévrovaciu schopnosť robota a tiež jeho prejazdnosť v teréne. Toto riešenie je komplikovanejšie nielen z pohľadu mechanickej konštrukcie, ale aj riadenie a elektronika takto riadeného robota je o niečo zložitejšia.



Obrázok 2.8: otočné koliesko

zdroj: <http://www.kolieska.sk/page.php?sekcia=produkty&idProdukt=5472>

2.3.4 Viackolesové podvozky

Tieto roboty tvoria početnú skupinu robotov. Často sa používajú hlavne pri nasadení v teréne. Majú vysokú statickú i dynamickú stabilitu. Spôsob riadenia je podobný ako pri predchádzajúcej skupine robotov (riadenými nápravami alebo diferencne). Výhodou je schopnosť zdolávať veľké terénne nerovnosti. Pri použití kĺbového podvozku sú tieto roboty schopné zdolávať aj schody. 6 kolesové robotické podvozky boli tiež použité aj pri viacerých robotických sondách na iných planétach [16].

2.3.5 Diferenčne riadené podvozky

Mobilné roboty s týmto hnacím systémom majú nezávislé poháňané dve kolesá a vpredu alebo vzadu oporné koleso či kolesá (obrázok 2.8). Riadenie pohybu je uskutočňované rozdielnou rýchlosťou prípadne aj smerom otáčania hnaných kolies. Toto riešenie umožňuje robotovi otočiť sa na mieste okolo osi nachádzajúcej sa v strede medzi dvoma hnanými kolesami. To je možné s výhodou využiť v uzavretom priestore a v kútoch.[2] Vďaka tejto vlastnosti sú to obľúbené roboty pre vnútorné použitie. Diferenčné riadenie sa tiež zvykne používať pri viackolesových a pásových podvozoch. Tie sú naopak obľúbené pre vonkajšie použitie.

Veľkým problémom tretieho, voľného kolesa je fakt že každé otočné koliesko má istú takzvanú „mŕtvu“ polohu. Je to poloha, pri ktorej sa smer, ktorým koliesko nie je schopné ísť, zhoduje s požadovaným smerom jazdy robota. Vtedy dochádza k nepríjemnému trhnutiu robota v prípade veľkého výkonu hnacích motorov, alebo k zastaveniu robota v prípade malého výkonu hnacích motorov. Tento problém je možné riešiť niekoľkými spôsobmi.

- minimalizovať šírku mŕtvej polohy zúžením kontaktnej plochy kolesa a zeme
- softvérovo sa vyhýbať mŕtvej polohe kolieska
- použiť koliesko s kvalitnými ložiskami a tým zúžiť šírku mŕtvej polohy

- použiť všesmerové koliesko (obrázok 2.10) namiesto otočného kolieska
- použiť všesmerovú guľôčku (obrázok 2.9) namiesto otočného kolieska



Obrázok 2.9: všesmerová guľôčka
zdroj: <http://www.omnitrack.co.uk/>



Obrázok 2.10: všesmerové koliesko
zdroj: <http://www.acroname.com/robotics/parts/R76-4CM-ROLLER.html>

2.4 Riadenie

Riadenie robota je možné realizovať lokálne alebo vzdialene.

Lokálne riadenie je riadenie pri, ktorom je jeden robot riadený samostatnou riadiacou jednotkou, v ktorej má uložené informácie o svete a o svojej polohe v tomto svete. Táto jednotka je schopná ovládať motory robota a je schopná prijímať dáta zo senzorov robota. Nevýhodou takéhoto riešenia je fakt, že robot má informácie len o sebe a nemá žiadne informácie o ďalších robotoch vo svete. Toto riešenie je možné použiť pokiaľ predpokladáme, že budeme riadiť len jedného robota. Alebo pokiaľ bude mať robot schopnosť pomocou senzorov detekovať výskyt iných robotov na trati a vyhnúť sa kolízi s nimi.

Vzdialené riadenie je riadenie, pri ktorom sú viaceré roboty riadené pomocou jednej centrálnej riadiacej jednotky, ktorá má informácie o svete a o polohe jednotlivých robotov v tomto svete. Požiadavky na presun zasiela jednotlivým robotom, ktoré majú viac či menej autonómne riadiace jednotky. Tieto riadia lokálne funkcie robota (napríklad presun od jednej križovatky k nasledujúcej)

Kvôli jednoduchosti riešenia predpokladajme, že v našom svete sa bude nachádzať len jeden robot a preto nie je potrebné mať centrálnu riadiacu jednotku. Riadiaca jednotka robota bude niesť informácie o svete a polohe robota v ňom a bude súčasne ovládať robota (otáčať kolesami a čítať dáta zo senzorov)

Riadiacu jednotku robota je možné riešiť niekoľkými spôsobmi. Najjednoduchšou a pre zadaného robota úplne dostačujúcou riadiacou jednotou je mikrokontrolér. Dnešné mikrokontroléry sú značne výkonné, majú dostatočné množstvo vstupných a výstupných portov na riadenie jednoduchého robota. Nevýhodou je komplikované ladenie programu.

2.4.1 FITkit

FITkit je školská vývojová platforma. Výhodou riešenia pomocou platformy FITkit je, že okrem výkonného mikrokontroléru má aj FPGA. V FPGA by sa dali naprogramovať niektoré časti riadenia robota, napríklad stavové automaty riadenia motorov a čítania údajov zo senzorov. V mikrokontroléri by bolo možné naprogramovať navigačný softvér robota. Nevýhodou tohto riešenia je zdĺhavejšie ladenie, ako pri použití počítača a komplikovanejší návrh ovládania pomocou vstavanej klávesnice.

2.4.2 Pda

Vreckové počítače majú dostatočný výkon, skvelé možnosti navrhnutia užívateľského rozhrania, množstvo dostupných knižníc pre rôzne programovacie jazyky. Výhodou sú tiež rôzne možnosti bezdrôtovej komunikácie s okolím (wifi, bt, irda, gprs, ...). Nevýhodou sú neštandardné porty, ku ktorým by bolo potrebné pripojiť robota. Ďalšou výhodou je existencia operačného systému. Ten umožňuje prácu so súborami, čo by bolo vhodné na ukladanie mapy sveta. Nevýhodou je tiež pomerne zdĺhavé ladenie programu.

2.4.3 PC

Osobný počítač má najväčšiu výhodu v oblasti ladenia softvéru pre robota. Program môžeme písať, prekladať a spúšťať na tom istom zariadení, čo značne šetrí čas. Nevýhodou je veľkosť spotreba a hmotnosť osobného počítača. Tieto faktory neumožňujú umiestnenie tejto riadiacej jednotky priamo na robota a preto by bolo potrebné pripojiť robota a počítač pomocou káblu, ktorý si robot bude „ťaháť“ za sebou. Počas vývoja je ale takéto obmedzenie pomerne nepodstatné. V osobnom počítači sa nám ponúka široká škála komunikačných rozhraní. Za všetky spomeniem LPT port.

Kapitola 3

Návrh a implementácia

V tejto kapitole sa budem venovať môjmu návrhu a implementácii jednoduchého pohyblivého robota. Rozhodol som sa pre konštrukciu trojkolesového robota s dvoma diferenčne riadenými prednými kolesami a s tretím zadným voľným otočným kolesom. Robot má pred prednými kolesami 4 snímače čiernej čiary založené na princípe odrazu infračerveného svetla. Snímače sú zoradené v jednej línii. Robot je riadený pomocou počítača, ku ktorému je pripojený pomocou LPT portu. Robot je k počítaču pripojený pomocou kábla. Snímače čiernej čiary sú napájané napätím 5 voltov a motory sú napájané napätím 12 voltov.

3.1 Pohonná jednotka

Pri výbere vhodnej pohonnej jednotky pre moje riešenie jednoduchého pohyblivého robota som mal niekoľko základných podmienok.

- pohonná jednotka musí byť jednoducho ovládateľná pomocou mikrokontroléru
- musí byť jednoducho ovládateľná pomocou počítača s operačným systémom linux
- musí byť schopná otáčania nízkymi otáčkami bez použitia ďalšej prevodovky
- rotor motoru musí byť použiteľný ako náprava
- motor sa musí dať presne natočiť kvôli jednoduchšiemu riadeniu a navigácii

3.1.1 Zvolený motor

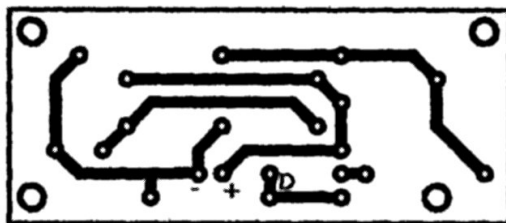
Po zhodnotení nasledujúcich podmienok som sa rozhodol pre krokový motorček zo starej 5,25" disketovej mechaniky (obrázok 2.3 na strane 8). Ide o motorček slúžiaci na vystavovanie čítacej hlavy. Tento motorček má kvôli presnosti rotor uložený v guľových ložiskách, čo dáva možnosť použiť ho ako nápravu. Motor vykoná 100 krokov na jednu obrátku a má 4 vinutia. Je možné budiť ho len unipolárne. Napájacie napätie motoru je 12 voltov.

3.1.2 Budenie motoru

Na uzemňovanie vinutí motoru som sa rozhodol použiť tranzistorové pole ULN2803. Vzhľadom k tomu že motor bol pôvodne určený k vystavovaniu čítacej hlavy je jeho krútiaci moment pomerne nízky. Preto som na budenie motoru použil unipolárne dvojfázové budenie.

3.2 Senzory

Robot ktorého som vyrábal má cestu, po ktorej sa bude pohybovať, vyznačenú pomocou čiernej čiary na svetlom podklade. V jazdnej dráhe robota sa nesmú nachádzať žiadne prekážky. Z tohto dôvodu nie je potrebné aby mal robot senzor detekujúci kolízie s prekážkami. Robot musí byť schopný detekovať čiernu čiaru na svetlom pozadí.



Obrázok 3.1: návrh plošného spoja

3.2.1 Zvolený senzor

Z cenových dôvodov som sa rozhodol pre vlastnoručne zhotovené snímače (obrázok 3.1). Pri návrhu snímača som z veľkej časti vychádzal zo schémy snímača na stránke <http://hw.cz/teorie-praxe/konstrukce/art2108-robot-v1-1.html>. Vzhľadom k tomu že snímače sú pripojené priamo k LPT portu, bolo nutné vykonať určité zmeny v zapojení tak, aby mal snímač stav vybud' 5 voltov, v prípade svetlého podkladu, alebo uzemnený výstup v prípade čierneho podkladu. Pôvodné zapojenie má stav bud' 5 voltov v prípade svetlého podkladu a nedefinovaný stav v prípade čierneho podkladu.

3.2.2 Rozloženie senzorov

Robot detekuje čiary nachádzajúce sa pod ním pomocou 4 senzorov umiestnených v rade pred prednými kolesami. Umiestnenie pred predné kolesa som volil z dôvodu, že takéto umiestnenie spôsobuje, že posun robotom doprava spôsobí aj posunutie senzorov doprava. Pri umiestnení za prednú nápravu by sa senzory pohybovali v opačnom smere ako je aktuálne otáčanie robota.

Senzory som si rozdelil na dve skupiny

Prvá skupina je dvojica senzorov uprostred, ktorá ma za úlohu detekovať hlavnú čiaru, po ktorej sa aktuálne robot pohybuje. Dvojica bola zvolená z toho dôvodu, že nám umožňuje detekovať mierne vybočenie zo správnej dráhy. To je realizované tak, že jeden zo senzorov opustí dráhu ako prvý, tým sa súčasne detekuje aj smer, ktorým bude potrebné vykonať korekciu. Robot je vďaka tomuto riešeniu schopný prechádzať aj zákrutami a znesie aj nepresné uloženie na dráhu.

Druhá skupina je dvojica senzorov jeden na pravom a jeden na ľavom okraji robota. Tieto senzory slúžia na detekovanie čiar, ktoré sú vedené kolmo na čiaru, po ktorej sa robot aktuálne pohybuje. Vďaka tomu je robot schopný detekovať križovatky. Senzory musia byť dostatočne ďaleko od stredu robota, inak by dochádzalo v prudkých zákrutách mylnému detekovaniu križovatky. V prípade veľmi prudkej zákruty, ale stále môže dochádzať k nesprávnej detekcii križovatky. Tento problém pri existujúcom rozložení pôjde odstrániť

len v prípade, ak by sme za križovatku považovali len štvorcestnú križovatku (všetky snímače detekujú čiaru)

3.3 Mechanika

Vzhľadom na vnútorné použitie robota som sa rozhodol pre trojkolesového diferenciálne riadeného robota. Hnané sú predné dve kolesá. Zadné tretie koleso je voľné, otočné. Riadenie je realizované rozdielnou rýchlosťou otáčania sa hnaných kolies, prípadne ich protichodným otáčaním. Predné kolesá sú vysústružené na mieru zo silonu. Sú uchytené priamo na rotor motoru. Motor je odspodu pripevnený k podvozku robota. Toto riešenie bolo zvolené z dôvodu konštrukčnej jednoduchosti.

3.3.1 Tretie voľne otočné koliesko

Pôvodne som plánoval ako tretí oporný bod robota použiť všesmerové koliesko alebo všesmerovú guľôčku. Žiaľ nikde v Českej Republike ani Slovenskej Republike sa mi ich nepodarilo zohnať a výroba by pravdepodobne bola príliš drahá. Všesmerové kolieska aj všesmerové guľôčky som vedel zohnať len z niekoľkých anglických internetových obchodov. Cena takéhoto kolieska bola žiaľ príliš vysoká.

Vzhľadom na nedostupnosť všesmerových koliesok a guľčiek som sa rozhodol použiť bežné otočné koliesko. Počas vývoja softvéru a testovania mi táto voľba spôsobila množstvo problémov. Koliesko sa často zasekávalo, niekedy nečakaným spôsobom zmenilo smer jazdy robota. Koliesko ktoré som použil malo značne nekvalitné ložiská a príliš širokú kontaktnú plochu so zemou. Kontaktnú plochu som oblepil úzkym prúžkom gumy a tým som zúžil kontaktnú plochu so zemou. Po následnom namazaní ložísk kolieska sa jeho chovanie výrazne zlepšilo. Stále však v niektorých polohách dochádza k jeho zablokovaniu a k zastaveniu robota.

3.4 Riadenie

Riadiacou jednotku môjho robota je počítač. Robot je pripojený pomocou LPT portu a je riadený programom napísaným v jazyku C. Programovací jazyk C bol zvolený z dôvodu jeho pomerne jednoduchšej prenositeľnosti na iné platformy ako je napríklad prenosný počítač, FITkit, alebo veľká časť mikrokontrolérov.

Ovládanie motorov a načítanie údajov zo snímača je riešené pomocou statickej knižnice *robotlib*. Navigácia robota po mape je v hlavnom programe *robot*.

3.4.1 Pripojovací kábel

Robot je k počítaču pripojený pomocou kábla. Týmto káblom je vedené aj napájanie. Pre snímače čiernej čiary je potrebných 12 liniek (3 linky pre každý snímač - napájanie, zem, dáta). Pre ovládanie motorov je privedených 10 liniek (4 dátové linky pre každý motor, spoločná zem a spoločné napájanie).

3.4.2 Knižnica robotlib

Pomocou knižnice *robotlib* je možné určiť pre každý motor zvlášť, či sa v aktuálnom kroku otočí dopredu, dozadu alebo či bude motor zastavený. Ďalej je možné pomocou knižnice

zistiť aktuálny stav čiar pod robotom (či je robot priamo na čiare, či detekoval križovatku, či potrebuje korekciu určitým smerom alebo či stratil čiaru). V tejto časti sa budem podrobnejšie venovať jednotlivým funkciám knižnice *robotlib*

Vymenované typy

V knižnici sú použité tri vymenované typy:

```
enum {  
    L, R, FW  
};
```

Tento vymenovaný typ slúži na určenie smeru odbočenia na križovatke.

```
enum {  
    LINE_OK, KOREKCIA_R, KOREKCIA_L, KRIZOVATKA, STRATA  
};
```

Tento vymenovaný typ slúži na určenie stavu detekovanej čiary.

```
enum {  
    OK=1,                // operácia skončila korektne  
    ERR_AKT_GT_STOP=-1,  // aktuálny stav automatu je väčší  
                        // ako povolená hodnota  
    ERR_AKT_LS_START=-2, // aktuálny stav automatu je menší  
                        // ako povolená hodnota  
    ERR_NEW_GT_STOP=-3,  // nasledujúci stav automatu je väčší  
                        // ako povolená hodnota  
    ERR_NEW_LS_START=-4, // nasledujúci stav automatu je menší  
                        // ako povolená hodnota  
    ERR_DIR=-10,         // chybné zvolený smer otáčania motoru  
    ERR_LPT_START=-11,   // nepodarilo sa získať prístup k portu LPT  
    ERR_LPT_STOP=-12,    // nepodarilo sa uvoľniť port LPT  
    ERR_LPT=-14          // zatiaľ nebol získaný prístup k portu LPT  
};
```

Tento vymenovaný typ obsahuje chybové stavy ktoré nám môžu vrátiť jednotlivé funkcie knižnice.

Štruktúry

Knižnica používa dve štruktúry na uchovávanie stavu a podávanie si tejto informácie medzi jednotlivými funkciami.

Prvá štruktúra uchováva momentálne len aktuálny stav čiar detekovaných senzormi. Bola zvolená forma uchovania v štruktúre kvôli možnému rozšíreniu do budúcnosti. Pôvodne som plánoval možnosť zapínania a vypínania snímačov podľa potreby a v tom prípade by v štruktúre pribudla informácia o tom, ktoré snímače sú aktuálne zapnuté a ktoré vypnuté.

```
typedef struct {  
    int state;  
} Status;
```

V druhej štruktúre sú uložené informácie o riadení krokových motorov. Alebo informácie o LPT porte a časoch čakania. V závislosti na čo sa aktuálne štruktúra používa. Táto štruktúra bola prevzatá z mojej knižnice *motorlib* s polu s ďalšími časťami. Táto pôvodná knižnica slúžila na ovládanie jediného krokového motoru pomocou LPT portu. Bola napísaná pre účely semestrálneho projektu. V pôvodnej verzii preto bol stav jediného motoru, nastavenia a stav LPT portu uložený spoločne. V tejto verzii sa v jednej premennej ukladá stav jedného motoru, v druhej stav druhého motoru a v tretej nastavenia a stav LPT portu.

```
typedef struct {
    int baseport; // adresa LPT portu
    int lpt;       // informácia o tom či je získaný prístup k LPT
    long int wait; // čas čakania medzi dvoma krokmi
    int startb;    // štartovací byt na ktorom je pripojený motor
    int stopb;     // koncový byt na ktorom je pripojený motor
    int state;     // aktuálny stav natočenia motoru
    int state2;    // aktuálny stav natočenia motoru pri dvojfázovom buzení
} Settings;
```

Užívateľsky prístupné funkcie

Funkcie pre inicializovanie štruktúr

```
void initLine(Status *st);
void init(Settings *s);
```

Funkcie pre nastavenie adresy LPT portu, doby čakania medzi dvoma krokmi, prvého a posledného bytu ku ktorému je pripojený krokový motor.

```
void SetBaseport(Settings *s, int port);
void SetWait(Settings *s, long int wait);
int SetStart(Settings *s, int start);
int SetStop(Settings *s, int stop);
```

Funkcie pre získanie práva prístupu k LPT a k uvoľneniu LPT portu. Funkciu pre získanie portu použijeme tesne po spustení programe, skôr ako sa ktorákoľvek funkcia pokúsi o prístup k portu LPT. V opačnom prípade by došlo k chybe. Tesne pred ukončením programu musíme port LPT uvoľniť, aby ho mohli poľa potreby využívať aj ostatné aplikácie.

```
int LPTstart(Settings *s);
int LPTstop(Settings *s);
```

Funkcia pre otočenie motoru o jeden krok. Funkcia má ako parametre ukazovateľ na štruktúru *s*, v ktorej sú uložené spoločné nastavenia od LPT portu. Ďalej dostane dva ukazovatele na štruktúry *s1* a *s2*, v ktorých sú uložené nastavenia a stavy prvého a druhého motoru. V premenných *direction 1* a *2* dostane funkcia informáciu o tom, ktorým smerom sa budú motory otáčať. Otáčanie je realizované pomocou dvojfázového budenia. Okrem toho je za účelom zvýšenia výkonu a zrovnomenenia chodu pridaný medzikrok, podobne ako je to pri otáčaní s polovičným krokom. Funkcia vykoná pol krok a následne v tesnom slede(experimentálne zistený vhodný čas, za ktorý motor ešte nestihne dokončiť polkrok) je vykonaný druhý polkrok, už bežný pre dvojfázové budenie motoru. Počas testovania robota sa táto kombinácia dvojfázového budenia s buzením s polovičným krokom ukázala ako najvhodnejšia.

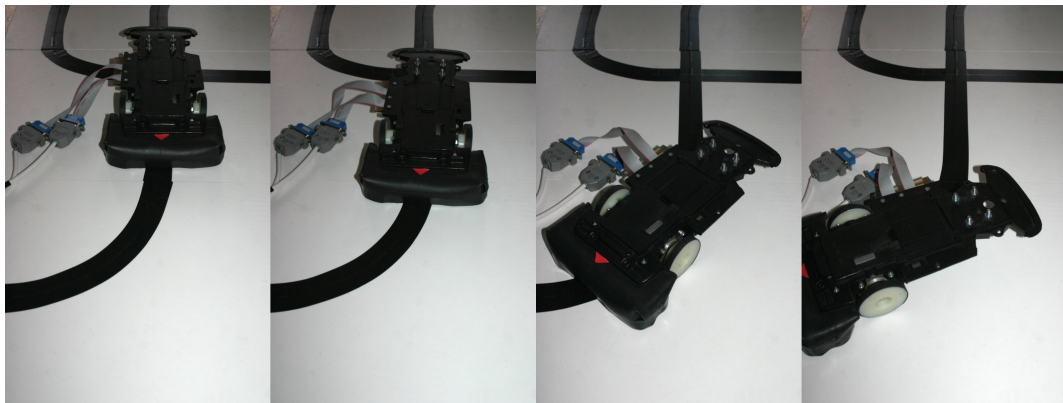

```
int Turn (Settings *s, Settings *s1, Settings *s2, int direction1, int direction2);
```

Funkcia Stop uvoľní kotvu motoru. Na LPT port pošle hodnotu 0 a tým sa zruší napájanie všetkých vinutí motoru. Funkcia sa používa pri ukončení programu robota, ale je možné použiť ju aj v prípade že predpokladáme dlhodobé státie robota na jednom mieste. Zabráni tak prehriatiu momentálne budeného vinutia.

```
int Stop (Settings *s);
```

Funkcia, ktorá je volaná v prípade, že bola detekovaná čiara. Funkcia robota len posunie o krok dopredu.

```
int ciara(Settings *s, Settings *s1, Settings *s2);
```



Obrázok 3.2: prejazd zákrutou

Tieto funkcie sú volané v prípade potreby korekcie smeru [3.2](#) robota doprava alebo doľava. Preberajú riadenie robota a vykoná definované kroky preupravu smeru. Riadenie odovzdajú hlavnému programu ktorý zisťuje, či sa čiara nachádza opäť pod dvojicou stredových senzorov. V prípade potreby sú funkcie volané opakovane. Funkcie je možné pozmeniť aj tak, že preveznú riadenie až kým nebude robot opäť v správnej polohe. Funkcie je možné upraviť aj tak, že spočiatku dochádza len k miernej korekcii a až neskôr k prudšiemu zatačaniu. Vzhľadom k prudkým zákrutám testovacej trate som zvolil verziu, v ktorej robot zatača od začiatku prudko.

```
int korekciaR(Settings *s, Settings *s1, Settings *s2);
```

```
int korekciaL(Settings *s, Settings *s1, Settings *s2);
```

Funkcia, ktorá je volaná v prípade detekovania križovatky [3.3](#). Na základe parametra turn na križovatke robot odbočí vpravo, vľavo, alebo pôjde rovno. Parameter turn je vypočítaný funkciou v samotnom programe.

```
int krizovatka(Settings *s, Settings *s1, Settings *s2, Status *st, int turn);
```

Funkcia, ktorá je volaná v prípade, že robot stratí čiaru. Momentálne spôsobí, že robot prejde asi 10 krokov dopredu a hľadá čiaru. Neskôr môže byť funkcia rozšírená na komplikovanejšie vyhľadávanie stratenej čiary.

```
int strata(Settings *s, Settings *s1, Settings *s2);
```



Obrázok 3.3: prejazd zkrížovatkou

3.4.3 Riadiaci program

Riadiaci program pozostáva z inicializácie štruktúr, uloženia mapy sveta do príslušných štruktúr a z nekonečnej slučky, v ktorej robot detekuje stav čiary a reaguje naň vykonávaním príslušných obslužných funkcií.

Mapa sveta

Mapa sveta je súčasťou programu, nenačítava sa z externého súboru. Toto riešenie bolo zvolené z dôvodu predpokladaného prenosu programu na platformu FitKit. Mapa sa ukladá do dvojice štruktúr.

Prvou je štruktúra križovatka

```
typedef struct {
    int cislo_krizovatky;
    int smery[4][10];
} Krizovatka;
```

V programe je vytvorené pole tejto štruktúry, kde každý záznam v poli reprezentuje jednu križovatku. V štruktúre je uložené číslo križovatky a ďalej pole smerov, ktoré má 4 riadky. V každom riadku sú uložené dáta k jednému smeru. V každom smere je 10 položkové pole intov. Prvých 8 pozícií poľa podáva informácie o cieľových staniciach dostupných v tomto konkrétnom smere. Posledné dve pozície udávajú číslo nasledujúcej križovatky a číslo smeru, z ktorého budeme do tejto križovatky vchádzať. To je dôležité pre výpočet, či budeme na križovatke odbáčať vpravo alebo vľavo alebo prejdeme križovatkou rovno.

Druhou štruktúrou je štruktúra stanovisko. Je to štruktúra zberajúca informácie o cieľových stanoviskách robota.

```
typedef struct {
    int cislo_krizovatky;
    int cislo_stanoviska;
    int smer;
} Stanovisko;
```

V programe je vytvorené pole tejto štruktúry, kde každý záznam v poli reprezentuje jedno koncové stanovisko robota. V štruktúre je uložené číslo koncového bodu, číslo križovatky,

z ktorej koncový bod vychádza a smer, z ktorého je koncové stanovisko pripojené na križovatku.

Funkcie `initStanovisko`, `initSmer`, `initKrižovatka` slúžia na uloženie hodnôt do štruktúr križovatka a stanovisko. Ako parametre očakáva funkcia ukazovateľ na príslušnú štruktúru a hodnoty, ktoré budeme zapisovať do štruktúry.

```
int initStanovisko(Stanovisko *stanovisko, int cs, int ck, int smer)
int initKrizovatka(Krizovatka *krizovatka, int ck)
int initSmer(Krizovatka *krizovatka, int smer,
int smer1,
int smer2,
int smer3,
int smer4,
int smer5,
int smer6,
int smer7,
int smer8,
int cislo_nasl_kr,
int smer_prichodu
)
```

Funkcia `turn` je určená na výpočet smeru odbočenia v aktuálnej križovatke. Funkcia dostane ako parametre ukazovateľ na aktuálnu križovatku a na smer, z ktorého do tejto križovatky vošiel. Okrem toho dostane číslo cieľového koncového bodu. Funkcia vracia informáciu o smere odbočenia (R,L,FW) alebo chybový stav. Funkcia na základe čísla smeru, z ktorého sme prišli a čísla smeru, na ktorý sa chceme dostať vypočíta, do ktorej strany sa máme na križovatke vybrať.

```
int turn(Krizovatka *krizovatka, int *smer, int ciel)
```

Funkcia `Pozicia` vypočíta na základe štruktúry nesúcej informácie o aktuálnej križovatke a informácie o cieľovom koncovom bode, informáciu o tom, aké číslo má nasledujúca križovatka (zistí novú polohu robota)

Funkcia `Smer` vypočíta na základe tých istých informácií ako predchádzajúca funkcia informáciu o tom, z ktorého smeru robot do nasledujúcej križovatky bude prichádzať.

```
int Pozicia(Krizovatka *krizovatka, int ciel)
int Smer(Krizovatka *krizovatka, int ciel)
```

Hlavná funkcia programu je nekonečný cyklus, v ktorom sa pri každom kroku detekuje stav čiar pod robotom a následne sa reaguje príslušnou obslužnou funkciou. Navigačný výpočet je súčasťou reakcie na nájdenie križovatky. Po vojení do križovatky robot skontroluje či, bola očakávaná nejaká križovatka (kontroluje sa nenulový stav `premennej` `pozicia` a `smer` ktoré udávajú, do akej križovatky a z akého smeru sa robot momentálne blíži) V prípade, že sa robot nemal dostať do žiadnej križovatky, (premenná `pozicia` alebo `smer` je nulová), program sa ukončí. V opačnom prípade sa pomocou funkcie `turn` vypočíta smer odbočenia v tejto križovatke. Ten sa uloží do premennej `turnX` a následne sa volá funkcia `križovatka`, ktorá realizuje prechod križovatkou zvoleným smerom. Prechod je realizovaný ako pevná sekvencia krokov. Počas prejazdu križovatkou nie je nijako kontrolovaný stav čiar pod robotom. Na záver sa vypočíta číslo a smer príchodu nasledujúcej križovatky.

Tvar trate medzi dvoma križovatkami nie je robotu nijak známy a prípadne zákruty sú detekované ako potreba korigovať jazdu určitým smerom.

```

int main(int argc, char *argv[]) {
    // v tejto časti programu sa inicializujú premenné
    // a do štruktúr sa načítava mapa sveta

    int start=atoi(argv[1]);
    int ciel=atoi(argv[2]);
    // načítanie parametrov štartu a cieľu robota

    int pozicia=stanoviska[start-1].cislo_krizovatky;
    int smer=stanoviska[start-1].smer;
    // určenie križovatky ku ktorej robot príde a smeru z ktorého príde

    int smerOld;
    int turnX;
    // pomocné premenné

    // nekonečný cyklus, hlavná časť programu
    while (1)
    {
        if (Line(&s, &st)==ERR_LPT) break;
        // načítanie stavu čiar pod robotom. v prípade chyby sa program ukončí
        switch (st.state)
        {
            case LINE_OK:
                err=ciara(&s, &s1, &s2);
                break;
            case STRATA:
                err=strata(&s, &s1, &s2);
                break;
            case KOREKCIA_R: korekciaR(&s, &s1, &s2); break;
            case KOREKCIA_L: korekciaL(&s, &s1, &s2); break;
            case KRIZOVATKA:
                if (pozicia<1 || smer <1) break;
                turnX=turn(&(krizovatky[pozicia-1]), &smer, ciel);
                krizovatka(&s, &s1, &s2, &st, turnX);
                smerOld=smer;
                smer=krizovatky[pozicia-1].smer[smerOld][9];
                pozicia=krizovatky[pozicia-1].smer[smerOld][8];
                break;
        }
        if (err<0) break;
        if (i>10000) break;
        i++;
    }
    if (Stop(&s)<0) perror("Nepodarilo zastavit");
    LPTstop(&s);
}

```

Kapitola 4

Testovanie

Testovanie robota prebiehalo priebežne počas jeho vývoja po jednotlivých častiach. Po skonštruovaní celkového robota bol nakoniec podrobený kompletným testom. Na základe výsledkov testov bol vždy upravovaný po konštrukčnej aj programovej stránke.

4.1 Priebežne testovanie

V priebežnom testovaní som sa postupne venoval jednotlivým častiam v poradí, v akom boli kompletované. Prvou vecou, ktorú som mohol testovať, bolo riadenie jedného krokového motoru, následne som dokončil podvozok a naň umiestnil oba motory a pridal kolieska. Nasledoval výber a umiestnenie tretieho kolieska, konštruovanie snímačov a ich testovanie. Umiestnenie snímačov na robota a testy spojené s rozložením snímačov.

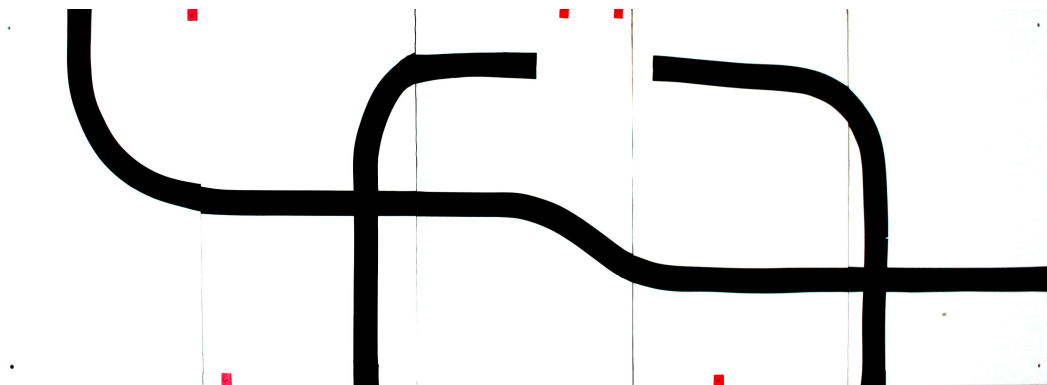
4.1.1 Testovanie riadenia jedného motoru

Po pripojení prvého motorčeka k portu LPT a zhotoveniu prvého jednoduchého programu, ktorý len pomocou jedného cyklu a bez stavového automatu otáčal motorom v jednom smere pomocou jednofázového unipolárneho budenia, prišli prvé testy. Výsledkom testov bolo zistenie, že motorček má pomerne malý krútiaci moment. Na rad prišla úvaha, že pokiaľ budem chcieť použiť motorček priamo bez prevodovky bude, potrebné použiť čo najmenšie kolieska a zvýšiť krútiaci moment pomocou unipolárneho dvojfázového budenia. Z tohto faktu som neskôr vychádzal aj pri návrhu knižnice na ovládanie motorčeka tak, aby bolo možné jednoducho pridať možnosť unipolárneho dvojfázového budenia. Okrem toho prišla myšlienka, že robota bude potrebné maximálne odľahčiť a preto bol pôvodne zamýšľaný oceľový podvozok nahradený plastovým.

4.1.2 Testovanie podvozku po osadení motorov a tretieho oporného kolieska

Motory a kolesá sú osadené zo spodnej časti podvozku a podvozok tak tvorí akúsi strechu robota. Toto riešenie bolo zvolené z dôvodu maximálneho zmenšovania koliesok kvôli nízkemu krútiacemu momentu.

Kolieska sú vysústružené zo silonu, ktorý sa hneď ukázal ako veľmi klzký materiál a robot prehrabával kolieskami na mieste. Preto boli kolieska navlečené do úzkeho prúžku gummy z bicyklovej duše. Problém s preklzavaním koliesok sa tým odstránil. Guma na



Obrázok 4.1: testovaica dráha

koliesku drží len vlastným trením a preto ju treba po dlhej jazde mierne napraviť, lebo v zákrutách sa trochu zvlieka.

Robot bol testovaný najskôr s jednofázovým unipolárnym buđením. To sa ukázalo, podľa predpokladov, ako nepoužiteľné pre rozbeh. Po rozbehnutí ale dokázalo robota udržať v pohybe. Táto vlastnosť by sa dala vyžiť pri pohone robota z batérie za účelom zníženia spotreby.

Dvojfázové budenie robota sa ukázalo ako dostatočne výkonné (robot okrem vlastnej hmotnosti dokáže doviesť približne 2kg záťaže). Problémom dvojfázového budenia bolo, že pri otáčaní kolies nízkymi otáčkami dochádzalo k značne trhanému chodu. To bolo odstránené použitím mierne upraveného unipolárneho budenia s polovičným krokom. Robot najskôr nabudí jedno vinutie ako pri buzení s polovičným krokom a tesne po tom nabudí aj druhé vinutie. Tieto dva kroky sa udejú ako reakcia na otočenie o jeden krok. Výhodou je jemnejší chod motorov pri nízkych otáčkach a zachovanie veľkého krútiaceho momentu dvojfázového budenia motoru.

4.1.3 Testovanie snímačov

Pri prvých testoch sa ukázalo že snímač reaguje veľmi dobre aj na slnečné svetlo a aj na umelé osvetlenie miestnosti. Preto bolo nutné snímače po namontovaní zatieniť tak aby na snímač dopadalo len svetlo z infradiód odrazené od snímaného povrchu.

Po viacerých testoch rozloženia snímačov som zvolil rozloženie v jednej línii. Toto rozloženie ponúka jednoduchú možnosť detekovania križovatiek bez nutnosti ukladať si informácie o predchádzajúcich stavoch ako by to bolo pri posunutí okrajových alebo stredových snímačov dopredu.

Snímač dokáže detekovať najlepšie čiaru širokú aspoň 1 cm a dlhú minimálne 2 cm. Keďže stredová vodiaca čiara je snímaná súčasne 2 snímačmi, ktoré sú umiestnené vedľa seba, je minimálna šírka tejto čiary 2,9 cm. Ja som pri testovaní používal čiaru vyrobenú nalepením dvoch pásov čiernej izolačnej pásky vedľa seba. Táto čiara mala šírku 3,2 cm. Križovatky musia byť kolmé aby bol robot schopný správne ich detekovať a následne aj prejsť. Nakoľko je šírka použitej čiary väčšia ako 2 cm, čo je minimálna dĺžka snímanej čiary, nemali snímače problém tieto kolmé čiary detekovať.

4.2 Testovanie kompletného robota

Po skompletovaní robota sa mohli začať testy celého robota. Tie by som rozdelil na dve fázy. Testovanie samotnej jazdy po jedinej čiare a testovanie s navigáciou [4.1](#).

4.2.1 Testovanie jednoduchej jazdy bez navigácie

Robota som testoval len na hladkom približne vodorovnom povrchu a v miernom stúpaní (do 10%).

Testovanie prejazdnosti

Vzhľadom k minimalizovaniu veľkosti koliesok sú niektoré časti robota veľmi blízko podkladu a preto je úplne nepoužiteľný v teréne. Jeho motory majú výkon nedostatočný na jazdu do stúpania. Robot dokáže stúpať približne do 10% stúpania. Do tohto stúpania vynesie len svoju vlastnú hmotnosť bez riadiacej jednotky či batérie. Nie je schopný sa v tomto stúpaní rozbehnúť a po zastavení nesmie byť odpojený od napájania, inak sa samovoľne rozbehne dole. Robot má snímače umiestnené vpredu a sú zatienené čiernou gumou, ktorá kvôli dobrej účinnosti siaha až takmer po zem. Táto skutočnosť minimalizuje schopnosť robota vystúpiť na prekážku. Robot dokáže bez väčších problémov zdolávať len prekážky vysoké asi 3 mm smerom hore a 5 mm pri zjazde z prekážky.

Testovanie prejazdu zákrut

Problémom zákrut je, že ak je zákruta príliš ostrá, môže byť mylne detekovaná ako križovatka, lebo bude časť cesty zosnímaná ako kolmá odbočka v križovatke. Testovaním sa ukázalo, že zákruta, ktorá má 90° môže mať pri šírke cesty 3,2 cm minimálny polomer zatočenia 8 cm od stredu čiar. V prípade že zákruta má menej ako 90° sa táto hodnota znižuje až na 6 cm. 6 cm je minimálny polomer zatočenia pri algoritme zatáčania, ktorý je momentálne použitý.

Pôvodný algoritmus prechádzal zákruty tak, že čím bola zákruta dlhšia, tým ostrejšie do nej zatáčal. To ale pôsobilo problém, že pri ostrých 90° zákrutách na začiatku takmer nezatočil a o to skôr ju vyhodnotil ako križovatku. Pri tomto algoritme bol minimálny polomer zatočenia až 13 cm. Aktuálny prechod zákrutou je taký, že pokiaľ nie je jedným z dvojice stredových snímačov detekovaná stredová čiara, tak sa príslušný motor zastaví a druhý motor pokračuje v chode. Prechod zákruty je preto trochu trhaný. Robot zákrutou prechádza o niečo pomalšie ako rovným úsekom. Je to z dôvodu, že zadná časť robota bola pri väčšej rýchlosti občas vynesená von a robot stratil čiaru.

Testovanie prejazdu križoviek

Robot od začiatku testovania križovatky bezproblémovo detekoval. Zvolený tvar snímača detekuje križovatky veľmi jednoduchým algoritmom. O niečo väčší problém bol s riadením prejazdu križovatkou. Z podobných dôvodov ako u zákrut aj pri prejazde križovatkou robot zníži rýchlosť.

Prejazd rovno je riešený sekvenciou po sebe idúcich krokov vpred počas ktorých sa nijak nedetekujú čiary pod robotom. Robot sa tak dostane až za križovatku a pokračuje v bežnej jazde. Toto riešenie bolo oveľa jednoduchšie a aj úspešnejšie, ako pokus rozoznať presne koniec križovatky pomocou bočných senzorov. Táto možnosť často zlyhávala hlavne preto,

že oba senzory naraz nedetekujú odbočky ale jeden senzor ju detekuje skôr a druhý neskôr. Variant slepého prejazdu sa ukázal ako funkčný a úspešný.

Odbočenie v križovatke doprava alebo doľava je realizované tiež pomocou pevnej sekvencie krokov. Táto sekvencia bola zistená experimentálne. Problémom je, aby sa robot otočil presne o 90° . Niekedy, keď dôjde k zablokovaniu tretieho kolieska, alebo ak sa zasekne ovládací kábel tak sa robot nenatočí presne. Pokiaľ je po odbočení aspoň jeden z dvojice stredových snímačov na čiare, robot prekoná križovatku úspešne. Niekedy sa ale stane že sa niečo zasekne a robot sa otočí natoľko nepresne, že čiaru úplne stratí. Vtedy robot prejde desať krokov dopredu a ak ani potom nenarazí na čiaru, je program ukončený.

4.2.2 Testovanie s navigáciou

Navigácia po dokončení fungovala dobre. Robot bol testovaný na dráhe so 6 koncovými stanoviskami a s 2 križovatkami. Bol skúšaný prejazd z každého a do každého stanoviska a robot prešiel všetky kombinácie úspešne. Problém je, keď robot narazí na križovatku v čase, keď už žiadnu nečakal. Vtedy sa program ukončí. To sa môže stať napríklad v prípade, keď zadáme nesprávne číslo štartovacej pozície robota. Robotovi sme skúšali predstavovať tvar trate medzi dvoma križovatkami a nakoľko on prejdený tvar trate nepozná choval sa aj za takých okolností korektne. Táto vlastnosť je veľmi vhodná z toho dôvodu, že pokiaľ potrebujeme z nejakého dôvodu pozmeniť tvar trate medzi dvoma križovatkami, stačí pozmeniť značenie a ne je potrebný zásah do mapy.

Robot nie je schopný odhaliť nesprávne zadanie štartovej či cieľovej pozície a jeho chovanie je vtedy závislé od toho, či zadané pozície existujú alebo nie. Ak existujú tak sa bude v križovatkách chovať tak ako by to bola skutočne trať medzi žiadanými bodmi. Keď narazí na neočakávanú križovatku tak zastane a program sa ukončí. V prípade že štartovacia alebo cieľová pozícia neexistuje vôbec, robot zastane hneď na prvej križovatke.

4.2.3 Výsledky testovania

Robot je konštruovaný pre vnútorné použitie. Jeho schopnosti prejazdu terénu sú v podstate nulové. Pri návrhu trasy robota sme limitovaní jeho rozmermi: šírka 16 cm, dĺžka 28 cm, a jednak rozložením snímačov. Minimálna vzdialenosť dvoch súbežných čiar je 9 cm, inak by mohlo dochádzať k nesprávnemu detekovaniu križovatky. Minimálny polomer zatočenia pri hrúbke čiary 3,2 cm je 8 cm. Pred križovatkou je potrebný aspoň 4 cm dlhý rovný úsek aby robot vchádzal do križovatky správne natočený. Minimálna hrúbka čiar je 2,9 cm. Maximálna nosnosť robota je 2 kg pri použití súčasných motorov. Tieto parametre sú pre testovacie účely prototypu dostačujúce.

Kapitola 5

Záver

Výsledkom mojej práce je funkčný prototyp malého mobilného robota. Robot je schopný pohybovať sa vo svete pomocou navigácie založenej na sieti čiar. Robot si vo svojej riadiacej jednotke nesie informácie o topológii tejto siete a je tak schopný dostať sa zo štartovej pozície do cieľa. Robot ako som ho navrhol ja, by mohol dobre poslúžiť ako zmenšenina automatizovanej pohonnej jednotky na prepravu materiálu po fabrike.

5.1 praktické využitie

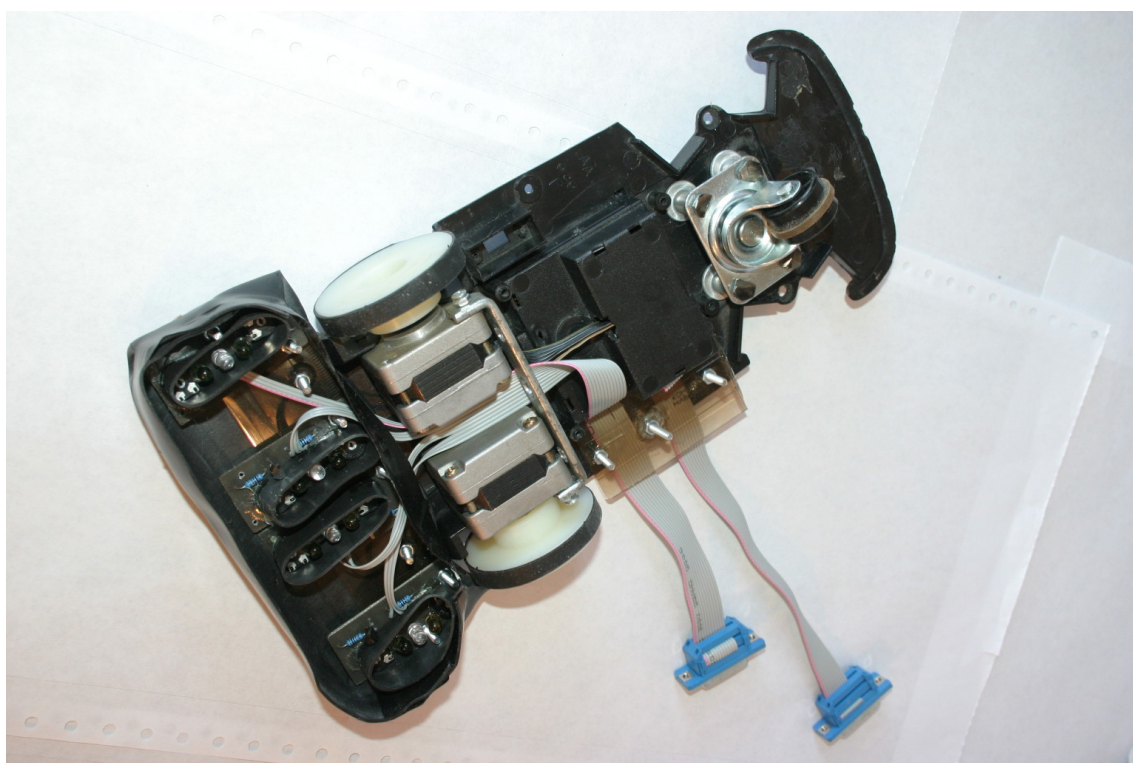
Robot použitý v skutočnej fabrike by sa vyrobil podstatne väčší a výkonnejší, aby bol schopný voziť veľké množstvá nákladu. Tiež by bolo potrebné vybaviť ho rôznymi senzormi, ktoré moje riešenie nezahrňuje. Šlo by napríklad o senzory včasne detekujúce živé a neživé prekážky v jazdnej dráhe, schopnosť zmerať šírku naloženého nákladu a vyhnúť sa prekážkam, o ktoré by mohol zavadiť náklad. Ďalej by bolo potrebné aby, bol robot schopný komunikovať s centrálou a koordinovať svoj pohyb s presunmi ostatných podobných robotov. Tiež by tento priemyslený robot mal byť schopný včas detekovať nedostatok energie v akumulátoroch a byť schopný si ich sám vymeniť za nabité akumulátory, tak, aby nemusel dlho stáť. Tiež by mohol byť schopný napríklad pomocou robotической ruky vykladať a nakladať tovar, ktorý nesie.

Literatúra

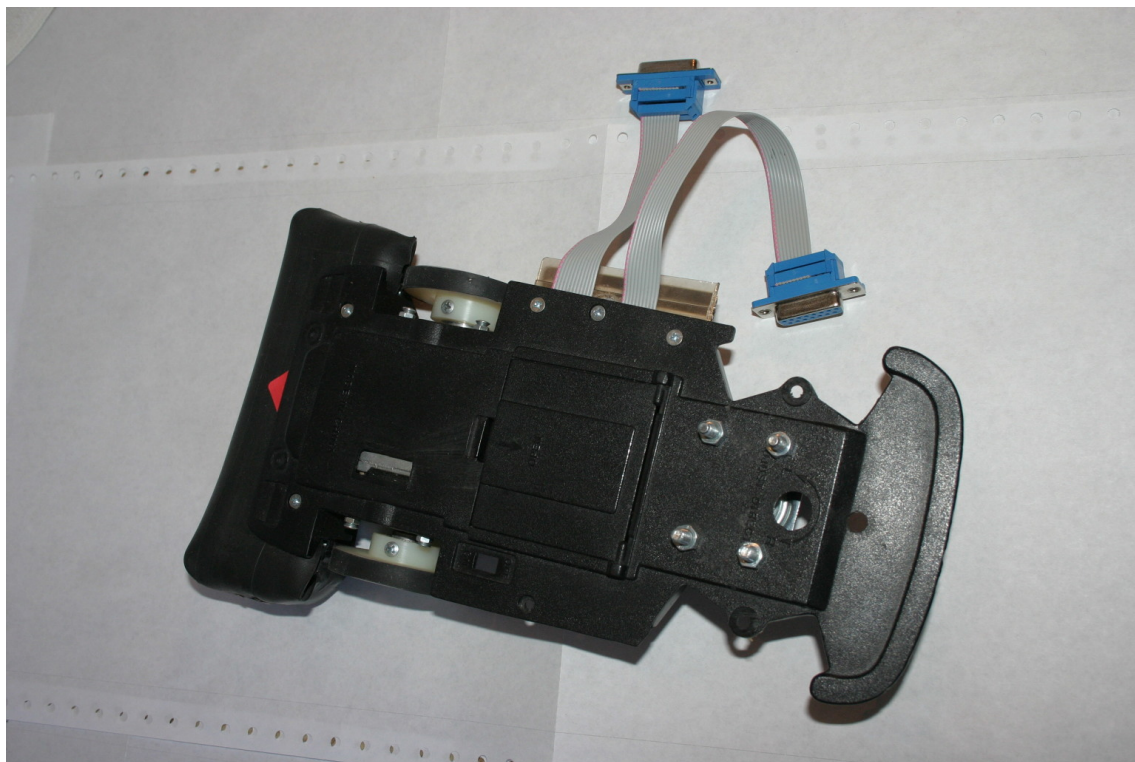
- [1] WWW stránky. ccd snímač čiary.
<http://www.robotika.sk/mains.php?page=/projects/sensorline/snimac.php>.
- [2] WWW stránky. Diferenčne riadené roboty.
http://robot.vsb.cz/elekskripta/servisni_roboty/interest1312221111.htm.
- [3] WWW stránky. Dvojkolesové mobilné roboty.
http://robot.vsb.cz/elekskripta/servisni_roboty/interest13122211.htm.
- [4] WWW stránky. Infra snímač.
<http://hw.cz/teorie-praxe/konstrukce/art2108-robot-v1-1.html>.
- [5] WWW stránky. Jedkokolesové mobilné roboty.
http://robot.vsb.cz/elekskripta/servisni_roboty/interest1312221.htm.
- [6] WWW stránky. Krokové motory.
http://robotika.yweb.sk/teoria/motory_2.html.
- [7] WWW stránky. Krokové motory. <http://robotika.cz/articles/steppers/cs>.
- [8] WWW stránky. Mobilné servisné roboty.
http://robot.vsb.cz/elekskripta/servisni_roboty/interest.htm.
- [9] WWW stránky. Premyselné optické snímače.
http://www.balluff.cz/bos_pouziti.asp.
- [10] WWW stránky. Robot - wikipédia. <http://sk.wikipedia.org/wiki/Robot>.
- [11] WWW stránky. Robotická kosačka.
<http://www.husqvarna.cz/node1556.aspx?pid=6720>.
- [12] WWW stránky. Senzor. <http://www.cudzieslova.sk/hladanie/senzor>.
- [13] WWW stránky. Servopohony.
<http://virtuallab.kar.elf.stuba.sk/robowiki/index.php/Servo/sk>.
- [14] WWW stránky. Svetlo. <http://cs.wikipedia.org/wiki/Sv%C4%9Btlo>.
- [15] WWW stránky. Trojkolesové a štvorkolesové mobilné roboty.
http://robot.vsb.cz/elekskripta/servisni_roboty/interest1312221111.htm.
- [16] WWW stránky. Viackolesové mobilné roboty.
http://robot.vsb.cz/elekskripta/servisni_roboty/interest1312221111111111.htm.

Dodatok A

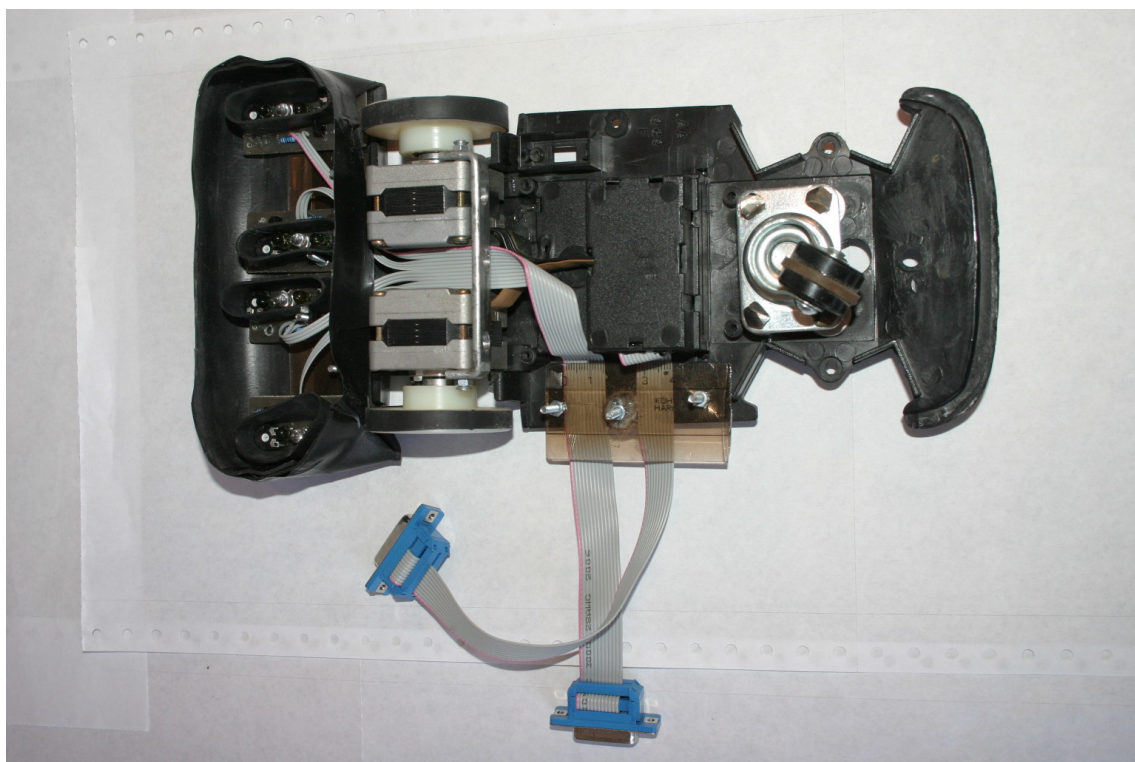
Fotografie robota



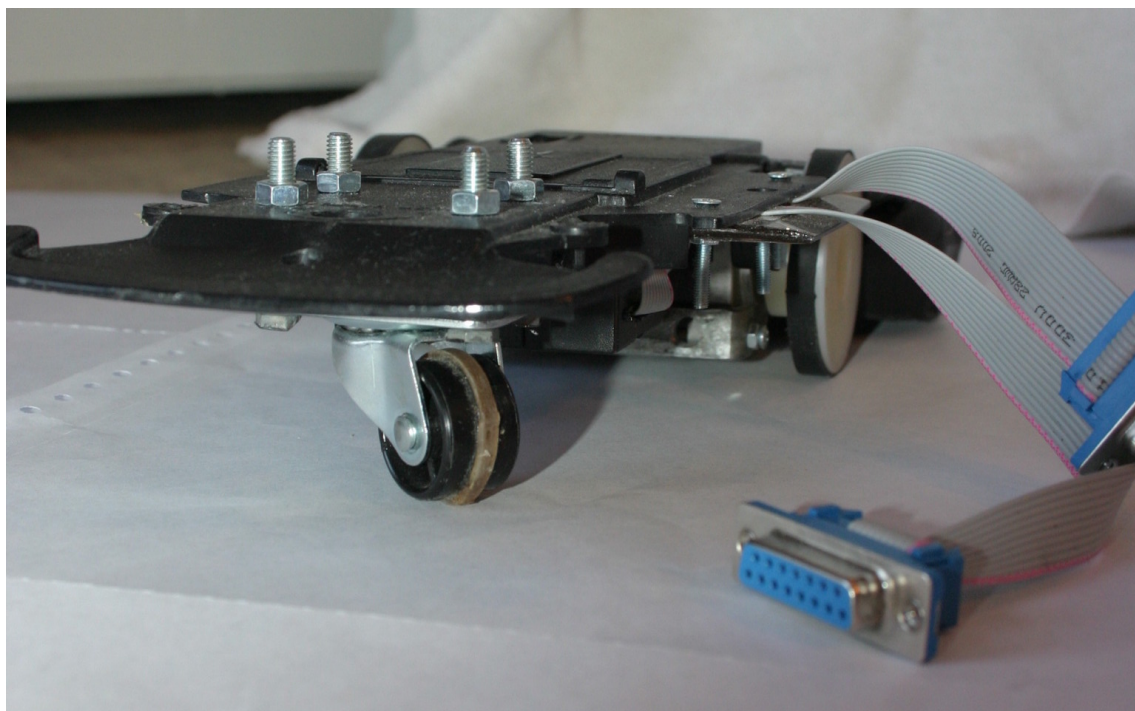
Obrázok A.1: Pohľad na robota zo spodnej strany. V prejdeji časti je vidieť rozloženie 4 snímačov a ich zatienenie



Obrázok A.2: pohľad z vrchu



Obrázok A.3: pohľad zo spodu



Obrázok A.4: pohľad zo zadu



Obrázok A.5: pohľad z boku